Package 'wk'

August 3, 2020

Title Lightweight Well-Known Geometry Parsing

Version 0.3.2

Maintainer Dewey Dunnington <dewey@fishandwhistle.net>

Description Provides a minimal R and C++ API for parsing well-known binary and well-known text representation of geometries to and from R-native formats. Well-known binary is compact and fast to parse; well-known text is human-readable and is useful for writing tests. These formats are only useful in R if the information they contain can be accessed in R, for which high-performance functions are provided here.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

LinkingTo Rcpp

Imports Rcpp

Suggests testthat, vctrs (>= 0.3.0), wkutils

URL https://paleolimbot.github.io/wk, https://github.com/paleolimbot/wk

BugReports https://github.com/paleolimbot/wk/issues

NeedsCompilation yes

Author Dewey Dunnington [aut, cre] (<https://orcid.org/0000-0002-9415-4582>), Edzer Pebesma [aut] (<https://orcid.org/0000-0001-8049-7069>)

Repository CRAN

Date/Publication 2020-08-03 10:20:02 UTC

R topics documented:

`new_wk_wkb` *S3 Details for wk_wkb*

Description

S3 Details for wk_wkb

Usage

```
new_wk_wkb(x = list())

validate_wk_wkb(x)

is_wk_wkb(x)
```

Arguments

x	A (possibly) `wkb()` vector

`new_wk_wksxp` *S3 Details for wk_wksxp*

Description

S3 Details for wk_wksxp

Usage

```
new_wk_wksxp(x = list())

is_wk_wksxp(x)

validate_wk_wksxp(x)
```

Arguments

x	A (possibly) wksxp() vector

`new_wk_wkt` *S3 Details for wk_wkt*

Description

S3 Details for wk_wkt

Usage

```
new_wk_wkt(x = character())

is_wk_wkt(x)

validate_wk_wkt(x)
```

Arguments

x	A (possibly) wkt() vector

`plot.wk_wkt` *Plot well-known geometry vectors*

Description

Plot well-known geometry vectors

Usage

```
## S3 method for class 'wk_wkt'
plot(
  x,
  ...,
  asp = 1,
  bbox = NULL,
  xlab = "",
  ylab = "",
  rule = "evenodd",
  add = FALSE
)

## S3 method for class 'wk_wkb'
plot(
  x,
  ...,
  asp = 1,
  bbox = NULL,
  xlab = "",
  ylab = "",
  rule = "evenodd",
  add = FALSE
)

## S3 method for class 'wk_wksxp'
plot(
  x,
  ...,
  asp = 1,
  bbox = NULL,
  xlab = "",
  ylab = "",
  rule = "evenodd",
  add = FALSE
)
```

Arguments

x	A wkt(), wkb(), or wksxp() vector.
...	Passed to plotting functions for features: graphics::points() for point and multipoint geometries, graphics::lines() for linestring and multilinestring geometries, and graphics::polypath() for polygon and multipolygon geometries.
asp	Passed to graphics::plot()
bbox	The limits of the plot in the form returned by wksxp_ranges().
xlab	Passed to graphics::plot()

ylab	Passed to graphics::plot()
rule	The rule to use for filling polygons (see graphics::polypath())
add	Should a new plot be created, or should x be added to the existing plot?

Value

The input, invisibly.

Examples

```
plot(as_wkt("LINESTRING (0 0, 1 1)"))
plot(as_wkb("LINESTRING (0 0, 1 1)"))
plot(as_wksxp("LINESTRING (0 0, 1 1)"))
```

---

vctrs-methods *Vctrs methods*

Description

Vctrs methods

Usage

```
vec_cast.wk_wkb(x, to, ...)

vec_ptype2.wk_wkb(x, y, ...)

vec_cast.wk_wkt(x, to, ...)

vec_ptype2.wk_wkt(x, y, ...)

vec_cast.wk_wksxp(x, to, ...)

vec_ptype2.wk_wksxp(x, y, ...)
```

Arguments

x, y, to, ...	See vctrs::vec_cast() and vctrs::vec_ptype2().

wkb *Mark lists of raw vectors as well-known binary*

Description

Mark lists of raw vectors as well-known binary

Usage

```
wkb(x = list())

parse_wkb(x)

as_wkb(x, ...)

## S3 method for class 'character'
as_wkb(x, ...)

## S3 method for class 'wk_wkb'
as_wkb(
  x,
  ...,
  include_z = NULL,
  include_m = NULL,
  include_srid = NULL,
  endian = NULL
)

## S3 method for class 'wk_wkt'
as_wkb(
  x,
  ...,
  include_z = NULL,
  include_m = NULL,
  include_srid = NULL,
  endian = NULL
)

## S3 method for class 'wk_wksxp'
as_wkb(
  x,
  ...,
  include_z = NULL,
  include_m = NULL,
  include_srid = NULL,
  endian = NULL
)
```

```
## S3 method for class 'blob'
as_wkb(x, ...)

## S3 method for class 'WKB'
as_wkb(x, ...)

## S3 method for class 'blob'
as_wksxp(x, ...)

## S3 method for class 'WKB'
as_wksxp(x, ...)
```

Arguments

x	A list() of raw() vectors or NULL.
...	Unused
include_z	Include the values of the Z and M coordinates and/or SRID in the output? Use FALSE to omit, TRUE to include, or NA to include only if present. Note that using TRUE may result in an error if there is no value present in the original.
include_m	Include the values of the Z and M coordinates and/or SRID in the output? Use FALSE to omit, TRUE to include, or NA to include only if present. Note that using TRUE may result in an error if there is no value present in the original.
include_srid	Include the values of the Z and M coordinates and/or SRID in the output? Use FALSE to omit, TRUE to include, or NA to include only if present. Note that using TRUE may result in an error if there is no value present in the original.
endian	For WKB writing, 0 for big endian, 1 for little endian. Defaults to wk_platform_endian() (slightly faster).

Value

A new_wk_wkb()

Examples

```
wkb(wkt_translate_wkb("POINT (20 10)"))
```

---

wkb_format *Format well-known geometry for printing*

Description

Provides an abbreviated version of the well-known text representation of a geometry. This returns a constant number of coordinates for each geometry, so is safe to use for geometry vectors with many (potentially large) features.

Usage

```
wkb_format(wkb, max_coords = 3)

wkt_format(wkt, max_coords = 3)

wksxp_format(wksxp, max_coords = 3)
```

Arguments

wkb	A list() of raw() vectors, such as that returned by sf::st_as_binary().
max_coords	The maximum number of coordinates to include in the output.
wkt	A character vector containing well-known text.
wksxp	A list() of classed objects

Value

A character vector of abbreviated well-known text.

Examples

```
wkt_format("MULTIPOLYGON (((0 0, 10 0, 0 10, 0 0)))")
wkb_format(
  wkt_translate_wkb(
    "MULTIPOLYGON (((0 0, 10 0, 0 10, 0 0)))"
  )
)
```

---

wkb_problems *Validate well-known binary and well-known text*

Description

Validate well-known binary and well-known text

Usage

```
wkb_problems(wkb)

wkt_problems(wkt)

wksxp_problems(wksxp)
```

Arguments

wkb	A list() of raw() vectors, such as that returned by sf::st_as_binary().
wkt	A character vector containing well-known text.
wksxp	A list() of classed objects

Value

A character vector of parsing errors. NA signifies that there was no parsing error.

Examples

```
# well-known text
wkt_problems(c("POINT EMTPY", "POINT (20 30)"))

# well-known binary
wkb <- wkt_translate_wkb("POINT (30 10)", endian = 1)[[1]]
wkb_bad <- wkb
wkb_bad[2] <- as.raw(255)
wkb_problems(list(wkb, wkb_bad))
```

---

wkb_translate_wkt *Translate between WKB and WKT*

Description

Translate between WKB and WKT

Usage

```
wkb_translate_wkt(
  wkb,
  include_z = NA,
  include_m = NA,
  include_srid = NA,
  precision = 16,
  trim = TRUE
)

wkb_translate_wkb(
  wkb,
  include_z = NA,
  include_m = NA,
  include_srid = NA,
  endian = wk_platform_endian(),
  buffer_size = 2048
)

wkb_translate_wksxp(wkb, include_z = NA, include_m = NA, include_srid = NA)

wkt_translate_wkt(
  wkt,
  include_z = NA,
```

```
  include_m = NA,
  include_srid = NA,
  precision = 16,
  trim = TRUE
)

wkt_translate_wkb(
  wkt,
  include_z = NA,
  include_m = NA,
  include_srid = NA,
  endian = wk_platform_endian(),
  buffer_size = 2048
)

wkt_translate_wksxp(wkt, include_z = NA, include_m = NA, include_srid = NA)

wksxp_translate_wkt(
  wksxp,
  include_z = NA,
  include_m = NA,
  include_srid = NA,
  precision = 16,
  trim = TRUE
)

wksxp_translate_wkb(
  wksxp,
  include_z = NA,
  include_m = NA,
  include_srid = NA,
  endian = wk_platform_endian(),
  buffer_size = 2048
)

wksxp_translate_wksxp(wksxp, include_z = NA, include_m = NA, include_srid = NA)

wk_platform_endian()
```

Arguments

wkb	A list() of raw() vectors, such as that returned by sf::st_as_binary().
include_z, include_m, include_srid	Include the values of the Z and M coordinates and/or SRID in the output? Use FALSE to omit, TRUE to include, or NA to include only if present. Note that using TRUE may result in an error if there is no value present in the original.
precision	The rounding precision to use when writing (number of decimal places).
trim	Trim unnecessary zeroes in the output?

endian	For WKB writing, 0 for big endian, 1 for little endian. Defaults to wk_platform_endian() (slightly faster).
buffer_size	For WKB writing, the initial buffer size to use for each feature, in bytes. This will be extended when needed, but if you are calling this repeatedly with huge geometries, setting this value to a larger number may result in less copying.
wkt	A character vector containing well-known text.
wksxp	A list() of classed objects

Value

*_translate_wkt() returns a character vector of well-known text; *_translate_wkb() returns a list of raw vectors, and *_translate_wksxp() returns an unclassed list of wksxp() geometries. Unlike as_wkb(), as_wkt(), and as_wksxp(), these functions do not attach a class to the output.

Examples

```
# translate between WKT and WKB
(wkb <- wkt_translate_wkb("POINT (30 10)"))
wkb_translate_wkt(wkb)

# some basic creation options are also available
wkt_translate_wkt("POINT (30 10)", trim = FALSE)
wkb_translate_wkb(wkb, endian = 0)
```

---

wksxp *Mark lists as well-known "S" expressions*

Description

Mark lists as well-known "S" expressions

Usage

```
wksxp(x = list())

parse_wksxp(x)

as_wksxp(x, ...)

## Default S3 method:
as_wksxp(x, ...)

## S3 method for class 'character'
as_wksxp(x, ...)

## S3 method for class 'wk_wksxp'
```

```
as_wksxp(x, ..., include_z = NULL, include_m = NULL, include_srid = NULL)

## S3 method for class 'wk_wkt'
as_wksxp(x, ..., include_z = NULL, include_m = NULL, include_srid = NULL)

## S3 method for class 'wk_wkb'
as_wksxp(x, ..., include_z = NULL, include_m = NULL, include_srid = NULL)
```

Arguments

x	A list() features (see details)
...	Unused
include_z	Include the values of the Z and M coordinates and/or SRID in the output? Use FALSE to omit, TRUE to include, or NA to include only if present. Note that using TRUE may result in an error if there is no value present in the original.
include_m	Include the values of the Z and M coordinates and/or SRID in the output? Use FALSE to omit, TRUE to include, or NA to include only if present. Note that using TRUE may result in an error if there is no value present in the original.
include_srid	Include the values of the Z and M coordinates and/or SRID in the output? Use FALSE to omit, TRUE to include, or NA to include only if present. Note that using TRUE may result in an error if there is no value present in the original.

Details

The "wksxp" format is experimental, but was written as a way to make it possible for packages to generate wkb() vectors without needing to use C++. The format represents geometries as following:

- points are matrices with zero or one row
- linestrings are matrices (one row per point)
- polygons are lists of matrices (one matrix per ring)
- multi (point, linestring, polygon) types are lists of the simple types (without any meta information)
- collections are lists of any type (must contain meta)

Any geometry that isn't in a multi type must have meta information encoded as attributes. The attribures that are used are:

- class: "wk_(pointllinestringl...)
- has_z: use TRUE if there is a Z coordinate (may be omitted if false)
- has_m: use TRUE if there is an M coordinate (may be omitted if false)

This is similar to the sf::st_sfc() format, but the formats aren't interchangable.

Value

A new_wk_wksxp()

Examples

```
wksxp(wkt_translate_wksxp("POINT (20 10)"))
```

---

wkt *Mark character vectors as well-known text*

Description

Mark character vectors as well-known text

Usage

```
wkt(x = character())

parse_wkt(x)

as_wkt(x, ...)

## Default S3 method:
as_wkt(x, ...)

## S3 method for class 'character'
as_wkt(x, ...)

## S3 method for class 'wk_wkt'
as_wkt(
  x,
  ...,
  include_z = NULL,
  include_m = NULL,
  include_srid = NULL,
  precision = NULL,
  trim = NULL
)

## S3 method for class 'wk_wkb'
as_wkt(
  x,
  ...,
  include_z = NULL,
  include_m = NULL,
  include_srid = NULL,
  precision = NULL,
  trim = NULL
)
```

```
## S3 method for class 'wk_wksxp'
as_wkt(
  x,
  ...,
  include_z = NULL,
  include_m = NULL,
  include_srid = NULL,
  precision = NULL,
  trim = NULL
)
```

Arguments

x	A character() vector containing well-known text.
...	Unused
include_z	Include the values of the Z and M coordinates and/or SRID in the output? Use FALSE to omit, TRUE to include, or NA to include only if present. Note that using TRUE may result in an error if there is no value present in the original.
include_m	Include the values of the Z and M coordinates and/or SRID in the output? Use FALSE to omit, TRUE to include, or NA to include only if present. Note that using TRUE may result in an error if there is no value present in the original.
include_srid	Include the values of the Z and M coordinates and/or SRID in the output? Use FALSE to omit, TRUE to include, or NA to include only if present. Note that using TRUE may result in an error if there is no value present in the original.
precision	The rounding precision to use when writing (number of decimal places).
trim	Trim unnecessary zeroes in the output?

Value

A new_wk_wkt()

Examples

```
wkt("POINT (20 10)")
```

Index